

УДК 681.3.006

M. Dolecki, Master of Science (Lublin Catholic University, Poland)**STATISTICAL ANALYSIS OF TREE PARITY MACHINE SYNCHRONIZATION TIME***

В статье рассмотрены особенности архитектуры и принципы взаимодействия двух искусственных нейронных сетей. Состояние синхронизации сетей позволяет использовать соответствующую информацию в качестве ключа для дальнейшего шифрования сообщений. С помощью специально разработанного программного средства собраны и обработаны обширные статистические данные о времени синхронизации сетей.

The article describes the features of architecture and the principles of interaction of two (artificial) neural networks. Synchronization status of networks allows to use the relevant information as a key to encrypt further communications. With the help of specially developed software tools the extensive statistical data on the networks synchronization time are collected and processed.

Introduction. Public-key cryptography is based on computationally hard problems, for which no efficient solving algorithm is known. Although efficient solving algorithm is not known, a development of computing power and the development of technologies allowing the cloud computing or the use of large botnet, allows to try to brute attacks. As the response to this development we observe constant increasing the length of keys used in cryptographic algorithm to the size which are considered as a safe one, but this also increases the time needed to encrypt and decrypt messages. An alternative approach is the search for a new methods of cryptography, which does not rely on computationally hard problems. Interesting are the attempts to apply artificial intelligence methods in cryptography, which review can be found in Ibrahim and Maarof [1]. One of the proposals is to use artificial neural networks to exchange the cryptographic keys for partner's further communication using open communication channel.

The research of the learning of artificial neural networks in the variant with the teacher led to the discovery of interesting phenomena which is the synchronization of these networks [2]. Network learning process requires the use of training set, which consists of pairs of input impulses – vector x and the expected response to the network, which can be single number or vector y . The network learning consists in the choice of weights, for which the network response will be the same as expected by the teacher, in other words learning process consists on minimization of network's error. A particularly interesting is possibility of modifying the training set during network's learning. In particular, the training set can be generated by another neural network. In this case, the learned network mimics teachers one, becoming at the end of learning a copy of teacher's network. However, if the networks will swap their roles and they will be for each other teacher and student, we will observe synchronization between them, after which both of the networks will have the same weight vectors. Synchronizing

networks have influence on each other by modifying the training set, so that the synchronization process is faster than passive learning of the network [3]. This difference in time required to achieve compatible weight vector by a third party is the foundation for the use of network synchronization in cryptography [4, 5]. In application are used a small neural networks with one hidden layer, which is called a TPM (Tree Parity Machine). Fig. 1 presents typical TPM, where K – number of perceptrons, N – number of input impulses for each perceptron and L the maximum absolute value of weight [6].

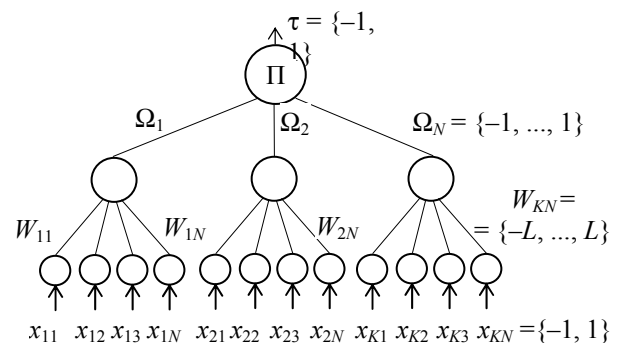


Fig. 1. Tree Parity Machine

Proposing a new approach to cryptographic key exchange resulted with attempts to attack synchronizing networks. The analysis of neural cryptography with three methods of a successful attack presented Kiimov, Mityagin and Shamir [7], who described the genetic, geometric and probabilistic attack. But as pointed out Ruttur [8], the probability of synchronization of attacker's network decreases exponentially with increasing values of weights, so with little cost we can achieve desired safety level. Another possibility of strengthening the safety of the cryptosystem based on synchronization of the networks is to expand the numbers field in which the TPMs operate to the Complex numbers, Quaternions and Octonions [9]. With these methods, the key exchange protocol using artificial neural networks is a relatively safe option.

*Статья подготовлена на основе сотрудничества между БГТУ и Люблинским католическим университетом (Польша).

Objectives. Two partners A and B during synchronization of their networks have access only to their own network's weight vector, actually generated vector of input impulses and the results of the both networks. Input impulses and results of the networks are transmitted with an open channel, while the weights vectors of the networks are known only to their owners, so partners does not know the weights of the other one network. Most essential problem is determining the point, at which both networks are already synchronized, that is, they have the same weight vectors. This is particularly important for communications safety. The more precisely you can specify this point of time, the faster you can finish network synchronization making it difficult to attack by a third party. Weight values obtained from the synchronization can be used as a cryptographic key for further communication. The purpose of this research was to analyze the synchronization of TPM with different parameters and specify the synchronization time and its dependence of the network structure. The time required to synchronize the network was defined by number of cycles, which consisted of: generating a input impulses, the calculating of output values for the network, sending the output to other networks and learning networks based on the common input impulses and output the second network. There were tested networks with different number of neurons and different number of input signals. In addition, the different weights intervals were analyzed.

Methodology. Analyses of neural nets synchronization was carried out on two networks running on one computer. Networks exchange only their results for the given input vector. As in synchronization with the open communication channel, none of the networks have access to the weight vector of the other network. The simulation program was checking after each step of synchronization if the weight vectors are identical and if they were, the simulation results were saved. Based on the synchronization results, statistical analyzes have been made using a spreadsheet.

Synchronization of TPM network on structures $K-N-L$, and $K \in [3; 10]$, $N \in [4; 20]$ and $L \in [5; 50]$ were simulated. Each pair of network was synchronized 1000 times. The results for a network of three neurons with four input impulses each are presented below. TPM on this structure is commonly used in neural cryptography.

Results. The simulations allowed to obtain for each network a similar histogram, which shows the number of networks that synchronized after a specified number of cycles. This histogram is similar to the histograms published in [2]. As can be seen most of the network synchronized in less than half of the longest observed time. The chart below (Fig. 2) shows the histogram for the network parameters 3-4-5 (3 perceptrons with 4 inputs each, weights belongs to the interval $[-5; 5]$).

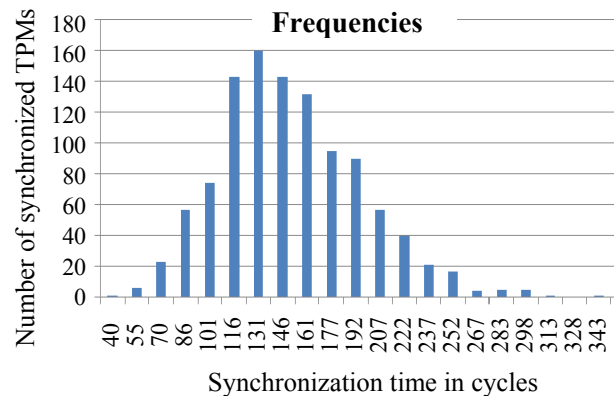


Fig. 2. Histogram presenting frequencies of synchronization of TPMs

For fixed number of perceptrons and the input impulses, the synchronization time increases proportional to the square of the maximum weight value, which is consistent with the Ruttor's et al. results [8]. The Table shows results of simulations for networks with three perceptrons, each with four input impulses, the weights value $L \in [5; 50]$, which corresponds to the system size $m = 2L + 1$ in the range 11 to 101.

Results of TPM synchronization

TPM structure	Average synchronization time	Standard deviation	Standard deviation (percentage)	Three quartile	Four quartile	Three quartile (percentage)
3-4-5	143,56	44,32	31%	172	328	52%
3-4-10	542,55	159,05	29%	635	1 283	49%
3-4-15	1 186,27	371,64	31%	1 386	3 714	37%
3-4-20	2 102,10	638,35	30%	2 475	4 830	51%
3-4-25	3 225,09	930,56	29%	3 740	7 501	50%
3-4-30	4 638,31	1 355,53	29%	5 416	11 181	48%
3-4-35	6 126,56	1 669,13	27%	7 136	13 979	51%
3-4-40	7 788,00	2 164,67	28%	9 077	19 763	46%
3-4-45	9 549,42	2 901,80	30%	10 818	26 228	41%
3-4-50	12 722,68	3 791,36	30%	14 921	34 561	43%

The table contains the average synchronization time measured in cycles, standard deviation and quartiles. As can be seen the deviation of the synchronization time is close to 30% and does not depend on the size m , but the most interesting result. It is the comparison three and four (maximum value) quartile. The average value of three quartile is about 50% of the longest observed synchronization time, which means that 75% of the network synchronizes in half the time than the worst of the analyzed network and it not depends on the weight maximum absolute value.

The Synchronization time in cycles dependence of the weights maximum absolute value grows with the square of the maximum weight value, as can be seen in the chart below.

The same consistency can be observed for networks with other structures.

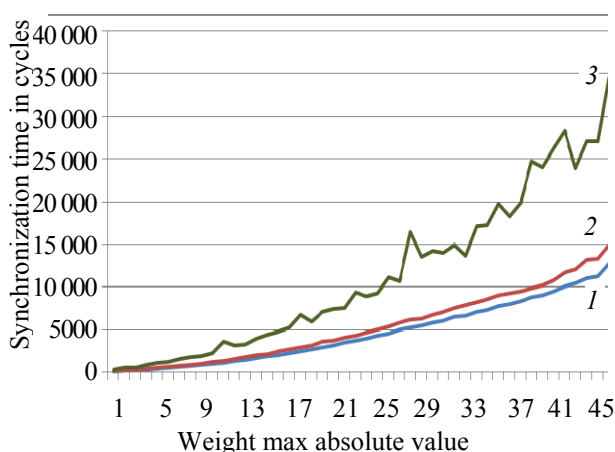


Fig. 3. Dependency of synchronization time on weight max absolute value:
1 – average; 2 – three quartile;
3 – maximal observed (four quartile)

Conclusions. The simulations shows that the average network synchronization time grows proportionally to the maximum weight value, while the standard deviation is about 30% of the average time.

Statistical analysis of the learning process indicates that the 75% of the network synchronizes in half the longest observed time. It follows that for any network there is a number of steps, which usually allows to synchronize networks, and when the synchronization process is continued for more steps it can be potentially danger by giving attacker

a longer time for a successful synchronization of his network. The number of steps, after which networks should be synchronization can be determined by the three quartile, (as v example for TPM with structure 3-4-10 this will be 635 cycles and for TPM 3-4-40 networks should be synchronized after 9077 cycles).

If the networks are not synchronized after this number of cycles, the process should be started all over again with random values of weights to improve safety of this system.

Bibliography

1. Ibrachim, S. A Review on Biological Inspired Computation in Cryptology / S. Ibrachim, M. Maarof // Jurnal Teknologi Maklumat. – 2005. – № 17 (1). – P. 90–98.
2. Kanter, I. Secure Exchange of information by synchronization of neural networks / I. Kanter, W. Kinzel, E. Kanter // Europhys. Lett. – 2002. – № 57. – P. 141–147.
3. Kinzel, W. Neural cryptography / W. Kinzel, I. Kanter // 9th International Conference on Neural Information Processing. – Singapore, 2002.
4. Kanter, I. The theory of neural networks and cryptography / I. Kanter, W. Kinzel // Proceedings of the XXII Solvay Conference on physics on the physics of communication. – 2002. – P. 631–644.
5. Synchronization of neural networks by mutual learning and its application to cryptography / E. Klein [et al.] // Advances in Neural Information Processing Systems. – 2005. – Vol. 17. – P. 689–696.
6. Volkmer, M. Tree Parity Machine Rekeying Architectures, Cryptology / M. Volkmer, S. Wallner // IEEE Transactions on Computers. – 2005. – Vol. 54, № 4. – P. 421–427.
7. Klimov, A. Analysis of Neural Cryptography / A. Klimov, A. Mityagin, A. Shamir // Advances in Cryptology – ASIACRYPT 2002. – 2003. – P. 288–298.
8. Genetic attack on neural cryptography / A. Ruttur [et al.] // Phys. Rev. E. – 2006. – № 73(3):036121.
9. Płonkowski, M. Algebraic aspects of mutual learning of neural networks / M. Płonkowski // New Electrical and Electronic Technologies and Their Industrial Implementation, Poland, June, 21–24. – Zakopane, 2005. – P. 125–127.

Поступила 02.03.2012